

A MODEL OF SOFTWARE PROCESS GENERATION OF A CONNECTIONIST SYSTEMS

Sonia I. Mariño

Departamento de Informática, Fac. de Ciencias Exactas y Naturales y Agrimensura
Universidad Nacional del Nordeste
Corrientes, Argentina
simarinio@yahoo.com

Abstract— In Computer Science there is a diversity of particular methods created for the development of products and services considering the disciplinary area they deal with. One of the areas of Computing is Artificial Intelligence. This, as a discipline, presents a variety of methods for the treatment and analysis of data in order to transform them into knowledge. Knowledge-based systems, as a sub-area of AI, expose models, methods and tools to simulate the behavior of domain specialists. The article exposes the adaptation of a method originally designed to construct symbolic systems in connectionist systems, differentiating themselves from other proposed ones. The conceptual framework, the method followed and the results are established. Finally, the conclusions are presented.

Keywords—Artificial Intelligence, connectionist systems, artificial neural networks, methods

I. INTRODUCCION

La Inteligencia Artificial (IA) como disciplina científica y tecnológica, proporciona fundamentos, métodos y herramientas para gestionar el conocimiento de los especialistas simulando su proceder ante procesos decisorios de diversa índole y complejidad.

Las Redes Neuronales Artificiales (RNA) son modelos comprendidos en el paradigma subsimbólico o conexionista de la IA y ampliamente aplicadas en diversas disciplinas. Lo expuesto ilustra el enfoque interdisciplinar, vastamente abordado en la actual sociedad del conocimiento, y que posibilita proponer y generar innovaciones para apoyar la toma de decisiones.

Los estudios iniciales permiten afirmar la inexistencia o la explicitación de una propuesta metodológica que integre en un método de la Ingeniería del Conocimiento, como es la metodología IDEAL, la creación de modelos de Redes Neuronales Artificiales.

El artículo presenta el proceso de investigación al que se asocia la elaboración de un modelo proceso software para construir modelos de RNA supervisadas. A fines de su validación el modelo propuesto se adapta al proceso de identificación botánica de especies.

A. Ingeniería del Conocimiento

La Ingeniería del Conocimiento —campo de la IA vinculada a la generación de sistemas inteligentes—, se define como el conjunto de principios, métodos y herramientas que permiten obtener el conocimiento desestructurado de los expertos, y aplicar el saber científico y la experiencia en la utilización de los conocimientos y de sus fuentes, mediante construcciones útiles para el hombre [1, 2]. Surge en la década de 1970 y en la actualidad se basa en métodos de la Computación y de la Información para representar el conocimiento y razonamiento humano en un determinado dominio en el marco de un sistema artificial.

Uno de los métodos comprendidos en la Ingeniería del Conocimiento es la metodología IDEAL [3], en que se sustenta la presente propuesta. Su nombre se funda en las fases que abarca: Identificación de la tarea, Desarrollo de los prototipos, Ejecución de la construcción del sistema integrado, Actuación para conseguir el mantenimiento perfecto y Lograr una adecuada transferencia tecnológica. Define un ciclo de vida del producto basado en el prototipaje rápido. Se cimenta en el modelo en espiral [4], en el que cada fase del ciclo de vida finaliza con el desarrollo de una versión o prototipo.

B. Sistemas conexionistas. Las Redes Neuronales Artificiales

Las redes neuronales o redes neuronales artificiales (RNA) o por sus siglas en inglés ANN (Artificial Neural Networks), puede definirse como un modelo matemático inspirado en el comportamiento biológico de las neuronas y en cómo se organizan formando la estructura del cerebro. Son redes densamente interconectadas de elementos computacionales simples llamados neuronas [7, 8].

Un modelo de RNA se diferencia de otros modelos de la IA, por el conocimiento implícito, éste se encuentra oculto en la estructura de la red y los pesos optimizados, entre los nodos. La Fig. 1 muestra un ejemplo de modelo neuronal con n entradas, y sus correspondencias con el modelo de neurona natural. Consta de los siguientes elementos:

- Un conjunto de entradas x_1, \dots, x_n
- Los pesos sinápticos w_1, \dots, w_n , correspondientes a cada entrada.
- Una función de agregación, Σ

- Una función de activación, f
- Una salida, Y .

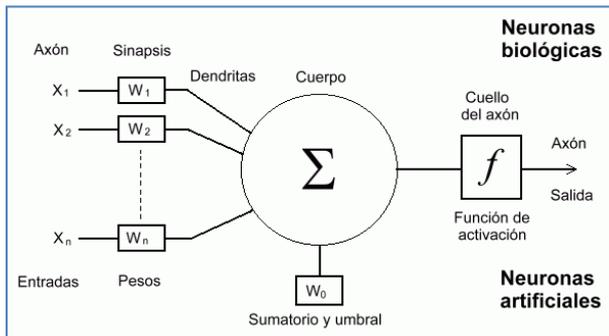


Fig. 1. Esquema de una neurona artificial y su analogía con una neurona natural (Fuente: otros).

Como generalidad se explica que las entradas de una RNA son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los parámetros libres del modelo, dado que pueden modificarse y ajustarse para realizar una tarea determinada.

Las neuronas de entrada (input) reciben los valores x_1, x_2, \dots, x_n , a partir del cual el modelo neuronal debe estimar un valor único. En este modelo, la salida neuronal Y se define por una función que se computa en cada neurona (1).

$$y = f\left(\sum_1^N wixi - T\right) \quad (1)$$

Donde:

w_i : pesos, almacenan el conocimiento generado en el modelo de RNA.

i : variables de entrada

T : valor umbral

Los resultados de algunas neuronas se conectan a las entradas de otras neuronas, conformando los denominados perceptrones multicapa o MLP. Un perceptrón multicapa se aplica para aproximar una función de clasificación que mapea el vector de entrada (x_1, x_2, \dots, x_n) a una o más clases G_1, G_2, \dots, G_m . Es decir, en el caso de estudio el vector de entrada representa las variables evidenciales, mientras que las categorías representan las variables objetivos.

La función de activación se elige de acuerdo a la tarea realizada por la neurona. Las más utilizadas en el aprendizaje del modelo son: identidad, escalón, sigmoidea, gaussiana, sinusoidal.

Las RNA presentan distintas topologías. En una topología MLP, las neuronas se agrupan en distintas capas (Fig. 2), siendo:

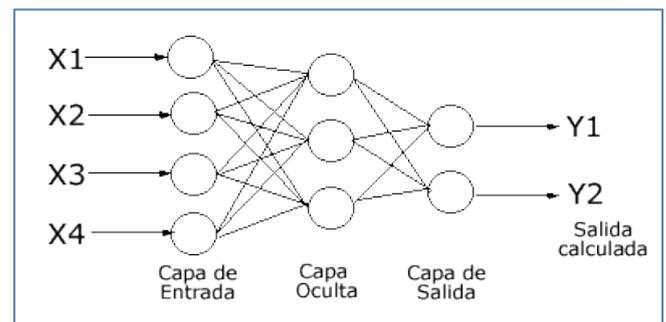
- capa de entrada formada por las entradas a la red,
- capa de salida formada por las neuronas que constituyen la salida final de la red, y
- capas ocultas formadas por las neuronas que se encuentran entre los nodos de entrada y de salida.

La salida de cada capa se conecta a la entrada de nodos en la siguiente capa. Las entradas de la primera capa (capa de entrada) son las entradas a la red, mientras que las salidas de la última capa forman la salida de la red.

Una RNA puede tener varias capas ocultas o no tener ninguna de ellas. Las conexiones sinápticas (las flechas que llegan y salen de las neuronas) indican el flujo de la señal a través de la red, y tienen asociadas un peso sináptico correspondiente. La topología de una RNA se define por el número de capas ocultas más la capa de salida.

Si la salida de una neurona se dirige hacia dos o más neuronas de la siguiente capa, cada una de estas últimas recibe la salida neta de la neurona anterior.

El problema habitual de las redes multicapa es, dado un conjunto de ejemplos clasificados y se conoce la salida deseada, proporcionar los pesos ajustados por la red para que se obtenga una aproximación correcta de las salidas si la red recibe



únicamente los datos de entrada.

Fig. 2. Topología de un Perceptrón multicapa (Adaptado de la literatura)

En la década de 1980 se diseñó el Algoritmo de Propagación hacia atrás (Backpropagation), que aproxima en numerosos casos los pesos a partir de los datos objetivo. La Fig. 3 muestra el algoritmo Backpropagation para el entrenamiento de la RNA

El problema habitual de las redes multicapa es, dado un conjunto de ejemplos clasificados y se conoce la salida deseada, proporcionar los pesos ajustados de la red para obtener una aproximación correcta de las salidas si la red recibe únicamente los datos de entrada.

En la década de 1980 se diseña el Algoritmo de Propagación hacia atrás (Backpropagation), que aproxima en numerosos casos los pesos a partir de los datos objetivo. La Fig. 3 muestra el algoritmo Backpropagation para el entrenamiento de la RNA

Inicializar los pesos sinápticos (generalmente aleatoriamente).

Introducir datos de entrada (en la capa de entrada) elegidos al azar entre el conjunto de datos de entrada usados para el entrenamiento.

Aplicar el método, la red genere un vector de datos de salida (propagación hacia adelante).

Comparar la salida generada por la red con la salida deseada (valor esperado de la variable objetivo).

Obtener el error, diferencia obtenida entre la salida generada y la deseada y utilizar para ajustar los pesos sinápticos de las neuronas de la capa de salidas.

Propagar el error hacia atrás, es decir, hacia la capa de neuronas anterior, (back-propagation). El error se usa para ajustar los pesos sinápticos en esta capa.

Continuar propagando el error hacia atrás y ajustar los pesos hasta alcanzar la capa de entradas.

Repetir el proceso con los diferentes datos de entrenamiento.

Fig. 3. Algoritmo Backpropagation para el entrenamiento de una RNA

II. MÉTODO

El método general abordado en el trabajo retoma el concepto de fases y no de etapas, siguiendo a [9]. Es decir, se entiende como una sucesión de momentos que pueden solaparse, consistiendo en:

- Revisión bibliográfica referente a metodologías de la Ingeniería del Conocimiento. Se opta por IDEAL [4].
- Revisión bibliográfica referente al desarrollo de los modelos supervisados de Redes Neuronales Artificiales, sus métodos y herramientas
- Adaptación de un modelo de proceso software descrito en [6] a la generación de modelos de RNA de apoyo a la toma de decisiones.

III. RESULTADOS

La propuesta que se expone, se fundamenta en que tanto los sistemas simbólicos como los sistemas conexionistas son sistemas basados en la experticia dado que representan el saber del especialista para

apoyar procesos de toma de decisiones de diversa complejidad. Por lo expuesto se considera valida la propuesta de integrar en un método de la Ingeniería del Conocimiento la construcción de modelos de RNA.

En referencia a la determinación de variables evidenciales relevantes y contribuyentes, los modelos conexionistas pueden entrenarse —proceso de aprendizaje automático— para entender el dominio del conocimiento, considerando el conjunto de variables evidenciales o los grupos relevantes de éstas seleccionadas por los referentes del dominio o por métodos automáticos. Las fases y actividades consisten en:

Fase I. Identificación de la tarea

Se ejecutan actividades similares a las descritas en [6] atinentes al proceso de determinación del dominio de conocimiento para la identificación de especies vegetales.

El proceso de la Fase I. Identificación de la tarea conlleva un conjunto de actividades concernientes a establecer: i) Plan de requerimientos y adquisición de conocimientos, ii) Evaluación y selección de las tareas, iii) Definición de las características de la tarea, centrado en determinar los requerimientos o necesidades de los EDC y los UF.

Se prioriza como técnica de recolección de información la entrevista, siendo el cuestionario y las observaciones expertas aplicadas a ambos perfiles de sujetos. Cabe aclarar que la heurística de los especialistas se incorpora en la explicitación de los archivos de datos o evidencias que posibilitarán la modelización y simulación computacional.

I.1. Plan de requerimientos y adquisición de conocimientos

Se determinan los objetivos del proyecto, se centran principalmente en la construcción del prototipo que aportaría a la toma de decisiones en problemas de identificación.

En esta fase se involucra a los EDC, potenciales usuarios de sistemas informáticos especializados en el dominio de conocimiento objeto de aplicación botánica. Es decir, un software inteligente construido para resolver o proporcionar una respuesta a un problema en particular. Dado que se trata de un sistema de apoyo en dominios de la Ciencia, es imprescindible la aplicación de los requisitos de fiabilidad y calidad, tanto de los conocimientos explicitados como de las respuestas brindadas ante una situación de identificación. Se entrevistan a especialistas en el dominio, y otros usuarios para corroborar la viabilidad y el aporte de introducir esta tecnología de la IA como complemento en procesos de identificación científica. Cabe aclarar que el conocimiento de los EDC para proponer una solución ante un problema de identificación se genera mayoritariamente en la experiencia y en un conjunto de heurísticas. Esta tarea se aborda en profundidad en la Fase II.

Se establece una vinculación entre la Ingeniería de Requerimientos y la Ingeniería del Conocimiento, dado

que el prototipo debe satisfacer las necesidades de los destinatarios. Es decir, las actividades que ambas contemplan deben enfocarse en obtener, analizar e implementar la visión de los distintos perfiles de usuarios que intervienen: Ingeniero de Conocimiento (IC), Experto en Dominio de Conocimiento (EDC) y Usuario Final (UF) (Fig. 4). La Experiencia de Usuario, desde la Interacción Humano-Computador, aportará en la definición de las interfaces aceptables por parte de los destinatarios.

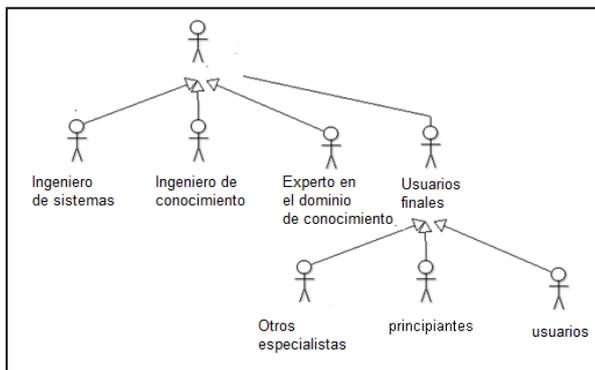


Fig. 4. Perfiles de sujetos intervinientes en el modelo de SCA (Fuente: [4])

I.2. Evaluación y selección de la tarea

Se obtiene información para definir el grado de dificultad de las tareas a desarrollar, es decir, el proceso de identificación mediado por tecnologías inteligentes. El estudio demuestra su complejidad, dado que el conocimiento disciplinar pertenece a los EDC, quienes a través de la observación y la puesta en escena de procesos inferenciales lo explicitan.

I.3 Definición de las características de la tarea

Esta fase involucra una descripción técnica completa formalizando las solicitudes de los EDC y UF. Se explicitan los requerimientos funcionales, operativos, de interfaz y de soporte. Entre los requerimientos no funcionales se destaca la definición de criterios de usabilidad en las interfaces hombre-máquina utilizadas como medio de comunicación.

Se definen escenarios de interacción entre el IC, el EDC y el UF vinculados a la construcción del SCA. El IC debe formular aquellas cuestiones orientadas a obtener los requerimientos de conocimientos para satisfacer a los EDC y a los UF. Los EDC son responsables de brindar el conocimiento para configurar la Base de Conocimientos, y se consulta a los UF sobre los problemas de conocimiento a resolver.

En síntesis, las actividades de la Fase I se orientan a determinar la viabilidad del proyecto, el que se considera factible dados los desarrollos de sistemas inteligentes para apoyar la toma de decisiones en dominios botánicos como los sintetizados en [3]. Se detecta la importancia de incorporar herramientas basadas en las tecnologías inteligentes y métodos automatizados para complementar el proceso de

identificación de especímenes vegetales. Por otra parte, se destaca que si una organización carece de especialistas para el reconocimiento de determinados ejemplares, se debe recurrir a otros expertos del mundo. Por lo expuesto, la construcción y difusión de sistemas inteligentes como el propuesto contribuyen con la tarea del experto del dominio, el taxónomo.

Fase II. Desarrollo de los prototipos

A continuación, se detallan las especificaciones de la Fase II, atinentes al modelo conexionista supervisado de razonamiento implementado:

II. 1 Concepción de la solución

Se evaluaron herramientas software para construir modelos conexionistas. Se generaron y validaron distintos modelos para simular la identificación de taxones.

II.2 Adquisición de conocimientos y conceptualización de los conocimientos.

Esta fase alcanza la adquisición y conceptualización de los conocimientos obtenidos de los EDC. Cada caso observado constituye un ejemplo que el modelo neuronal debe aprender. Se definen las variables evidenciales y los valores de la variable objetivo.

En los modelos supervisados se explicitan los valores que asume la variable objetivo, los valores que pueden asumir las variables o nodos de entrada. Además, se puede modelizar considerando las variables determinadas por el EDC o las relevantes seleccionadas automáticamente

II.3 Formalización de los conocimientos, comprende:

- Seleccionar los formalismos para representar computacionalmente los conocimientos.

- Completar la formalización de los conocimientos utilizando herramientas de un lenguaje de modelado de la Ingeniería de Requerimientos, según las funcionalidades definidas para el SCA.

- Definir los modelos neuronales artificiales para la identificación. Previamente se realizaron las siguientes actividades:

- Preprocesamiento de los datos. Los datos se preparan para lograr la construcción de modelos de RNA.
- Creación de los patrones de datos. Cada ejemplar, en el caso perteneciente al dominio, representa un patrón de datos. Se transforman los patrones a un archivo en el formato requerido para el procesamiento computacional. Lo expuesto, reflejó la transformación del conocimiento implícito en conocimiento explícito.

- Diseñar detalladamente el SCA conexionista.

- Definir los modelos de redes conexionistas para su entrenamiento, validación y testeo. Se

establecen las configuraciones propuestas según los parámetros que se establezcan.

- Determinar módulos que otorgan calidad, como el subsistema de explicación, las interfaces de comunicación con el UF y con otros sistemas informáticos, dado que éstos pueden considerarse como subsistemas comprendidos en un sistema informático de mayor alcance y complejidad.

II.4 Desarrollo del Sistema Cognitivo Artificial Conexionista

Se selecciona la herramienta computacional para simular la representación del conocimiento, y su posterior tratamiento con modelos conexionistas de la IA. Se distingue la construcción de los modelos de RNA y la programación de otros componentes del SCA que lo integran. Se abordan las siguientes actividades:

II.4.1 Construcción de modelos de RNA

Se realiza un estudio y revisión de conceptos relacionados con la topología, el dimensionamiento, la arquitectura y los algoritmos de aprendizaje de una RNA. El diseño y desarrollo de cada modelo de RNA propuesto implica:

- Definición de la Arquitectura. La arquitectura se define por el modelo de aprendizaje de la RNA, supervisado o no supervisado. En redes de aprendizaje supervisado se opta por un Perceptrón Multicapa (MLP). Las MLP son redes de alimentación hacia delante (feedforward), con una o más capas de unidades ocultas entre las unidades de entrada y las unidades de salida [7].

- Definición de archivos de datos. El conjunto de datos disponible se divide aleatoriamente en subconjuntos: entrenamiento, validación y testeo, manipulados considerando los criterios implementados por la herramienta computacional seleccionada.

- Creación y configuración de los modelos de RNA. Se crean distintos modelos de RNA, cada uno contiene tantos nodos de entrada como variables evidenciales y tantos nodos de salida como valores puede asumir la variable objetivo.

Para cada modelo de RNA se define la topología. Se proponen experimentalmente distintas configuraciones modificando el número de nodos y capas intermedias, funciones de activación y parámetros, debido a que se carece de un procedimiento específico para definir a priori cómo lograr un aprendizaje óptimo en modelos de RNA. Se consideran los parámetros de aprendizaje supervisado en MLP, algunos disponibles según las herramientas.

- Entrenamiento de la red. Se entrenan los modelos conexionistas utilizando el algoritmo de aprendizaje seleccionado. El proceso de entrenamiento consiste en el aprendizaje del modelo de conocimiento implícito en el conjunto de ejemplos representativos de un problema para detectar patrones.

El aprendizaje es un proceso que se repite n veces hasta minimizar el error global del modelo neuronal. La Fig. 3 ilustra el procedimiento sugerido para construir o entrenar modelos de RNA, una extensión del mismo se localiza en [7]:

- Evaluación y validación. Estas actividades aseguran que el modelo computacional sea una descripción adecuada del fenómeno en estudio, optándose por aquel que brinda una mejor respuesta ante un problema de conocimiento. Se procede a: calcular las medidas de calidad: en la evaluación y selección de los modelos se establecen como medidas que apoyan la decisión:

Una medida de calidad del modelo de aprendizaje se define en términos de los valores del error estándar como la suma de los cuadrados de los errores o SSQ y el error cuadrático medio o RMS

$$SSQ = \sum_{p=1}^R \|b_p - \bar{b}_p\|^2 \quad (2)$$

$$RMS = \sqrt{\sum_{p=1}^R \frac{\|b_p - \bar{b}_p\|^2}{r}} \quad (3)$$

Dónde:

\bar{b}_p es la salida de la red para el vector de entrada a_p ,

b_p es el valor de salida deseado para el vector de entrada a_p y r es el número de residuales.

El Porcentaje de Error (%E), indica el porcentaje de los ejemplos clasificados erróneamente. La Matriz de Confusión se utiliza para validar el reconocimiento de patrones.

- Comprobación. Entrenado el conjunto de datos, se procede a su verificación utilizando la técnica denominada Percent Split – que evalúa la calidad del clasificador según la calidad de un porcentaje de los datos reservados para testear el modelo— en la herramienta seleccionada.

Además, los modelos entrenados se validan, confrontándose los resultados obtenidos con los expertos del dominio o EDC. Si en el caso de estudio, se obtienen valores aceptables en las medidas de calidad estudiadas siendo los experimentos contrastados admitidos.

Las métricas, expresadas en (1) y (2), son indicadores de evaluación y selección de los modelos de RNA propuestos. Es decir, se valida su aplicación en procesos de aprendizaje previo a la clasificación de nuevos casos. Otros elementos de análisis son las representaciones gráficas que resumen el comportamiento del modelo RNA.

II.4.2 Desarrollo de interfaces e integración de modelos neuronales entrenados

En esta fase se ejecutan las siguientes actividades:

- Desarrollo de las interfaces y códigos de programación. Se crea la interfaz del usuario, para comunicar los resultados de la simulación. Ésta permite acceder a los modelos RNA supervisados entrenados y validados.
- Integración de los modelos de RNA. Los modelos elegidos para representar el dominio, se acceden desde las interfaces de usuario. Esta posibilidad permitirá asegurar la adopción del SCA por usuarios finales.

Actividades pertinentes a la Fase III Ejecución de la construcción del sistema integrado se pueden realizar iterativamente. La Fase IV Actuación para conseguir el mantenimiento y la Fase V Lograr una adecuada transferencia tecnológica se considera que conforman el contexto de aplicación, dado que ingresan a la escena otros sujetos, es decir, los destinatarios de estos sistemas.

IV. RESULTADOS

Se expuso un método adaptado desde la Inteligencia Artificial simbólica a la Inteligencia Artificial conexionista para modelar y simular el conocimiento de un dominio. Es así como se integran los modelos de Redes Neuronales Artificiales a un proceso software basado en el método IDEAL, diseñado originalmente para construir sistemas expertos, representante del paradigma simbólico de la IA.

Se entiende importante destacar que los artefactos software construidos aplicando el método expuesto, son propuestas rebatibles de la IA. Es decir, el conocimiento que encapsulan puede ser modificado ante nuevas evidencias que surgen después de construir, validar y aplicar el modelo de RNA.

Cabe destacar que la propuesta metodológica adopta la visión dialéctica, dado que el uso del artefacto producirá datos que sustentarán su evolución. Es así como la aplicación de acciones

orientadas al mantenimiento correctivo y perfecto en sucesivos ciclos de una espiral de conocimiento resulta de la tensión tesis-antítesis-síntesis como se evidencia en esta propuesta metodológica desde la Inteligencia Artificial.

REFERENCIAS

- [1] R. A. Aguilar Vera, J. C. Díaz Mendoza, G. E. Gómez Cruz, & E. L. Bojórquez (Eds), Ingeniería de Software e Ingeniería del Conocimiento: Tendencias de Investigación e Innovación Tecnológica en Iberoamérica, Alfaomega Grupo Editor S. A, 2010
- [2] P. Britos, P. Procesos de Explotación de Información Basados en Sistemas Inteligentes, Tesis Presentada para obtener el grado de Doctor en Ciencias Informáticas. Universidad Nacional de La Plata, 2008.
- [3] A. Gómez, N. Juristo, C. Montes & J. Pazos, Ingeniería del conocimiento. Ed. CEURA, 1997.
- [4] B. Boehm, "A Spiral Model of Software Development and Enhancement", IEEE P. Computer, IEEE, 21(5), pp. 61-72, 1998.
- [5] S. I. Mariño & M. Dematteis, Revisión de soluciones de tecnológicas inteligentes en biología, *Télématique* 13 (1), 2014, [en línea]: [Fecha de consulta: 7 de marzo de 2019] Disponible en: <<http://www.redalyc.org/articulo.oa?id=78429590003>> ISSN 1856-4194
- [6] S. I. Mariño, Modelo de gestión del conocimiento, Inédito, Universidad Nacional del Nordeste, 2018.
- [7] S. J. Russell & P. Norving, P. Inteligencia Artificial. Un enfoque práctico. Ed. Mc Graw Hill, 2004.
- [8] E. J. Rzepoluck, Neural network data analysis using Simulnet. Ed. Springer, 1998.
- [9] J. Samaja, Epistemología y Metodología. Elementos para una teoría de la Investigación Científica. Ed. EUDEBA, 2003.